Regular Paper

# Lightweight and Secure Certificateless Multi-receiver Encryption based on ECC

Ei Khaing Win[1,a) ] Tomoki Yoshihisa[1,b)] Yoshimasa Ishi[1,c)]
Tomoya Kawakami[2,d)] Yuuichi Teranishi[3,e)] Shinji Shimojo[1,f)]

Received: November 24, 2017, Accepted: June 8, 2018

***Abstract:*** In this paper, we propose an elliptic curve cryptography (ECC)-based certificateless multi-receiver encryption scheme for device to device communications on Internet of Things (IoT) applications. The proposed scheme eliminates computation expensive pairing operations to provide a lightweight multi-receiver encryption scheme, which has favourable properties for IoT applications. In addition to less time usage for both sender and receiver, the proposed scheme offers the necessary security properties such as source authentication, implicit user authentication, message integrity, and replay attack prevention for secure data exchange. In this paper, we show security proof for the proposed scheme based on the Elliptic Curve Discrete Logarithm Problem (ECDLP). We implemented our proposed scheme on a real embedded Android device and confirmed that it achieves less time cost for both encryption and decryption compared with the existing most efficient certificate-based multi-receiver encryption scheme and certificateless multi-receiver encryption scheme.

***Keywords:*** elliptic curve cryptography, authentication, certificateless, multi-receiver encryption, Internet of Things (IoT)

## 1. Introduction

Nowadays, the role of Internet of Things (IoT) grows in popularity as the number of smart devices and sensors increases. In some IoT applications related to healthcare, smart homes and group communications, sensitive data is exchanged among multiple users. For example, pedestrians who feel concerns for their health around a station send vital data, which is generated by body sensors and attached to smartphones, to nearby doctors or nurses with the purpose of improving their well-being, getting advice or healthcare. In such applications, a lightweight multi-receiver encryption scheme to ensure confidentiality, integrity, and authenticity is necessary to provide not only security but also efficiency.

To securely exchange information with fast computing speed, a symmetric encryption scheme can be used. However, it introduces security requirements such as authentication of parties in key agreement protocol or secure and integrity-assured key distribution to prevent man-in-the-middle attacks and other attacks. Without the use of public-key cryptography, a symmetric key scheme is not sufficient to get secure communication features such as confidentiality, authentication, integrity, and non-repudiation. Although the conventional public key infrastructure (PKI) is widely used in the current ICT systems, it is not suitable for resource-constrained devices due to its certificate overhead [1]. To avoid the certificate management problem, in identity-based cryptography introduced by Shamir [2], unique strings such as identities are used as public keys. However, an unconditionally trusted third party called key generation center (KGC) or (PKG) exists to generate system parameters and private keys for all users. As a result, a private key generator can eavesdrop all exchanged messages as it knows all users' private keys. This problem is called the key escrow problem. To simplify the certificate management of traditional public key cryptography and the key escrow problem of identity-based public key encryption, the concept of certificateless public key cryptography has been proposed by Al-Riyami and Paterson [3]. In certificateless public key cryptography (CL-PKE), the key generation process is split between PKG and the users. Key Generation Center issues the partial public key and the user generates their own public key and private key pair. To decrypt the ciphertext, both partial private key and private key are required. Knowing only partial private key, KGC cannot eavesdrop exchanged messages. In this way, the key escrow problem is avoided. Moreover, certificateless public key encryption eliminates the use of a certificate as the public key of the user is generated using the parameters given by KGC.

In devices with sensitive personal data, no trust exists on others but itself. Full control is in the hands of the device owner. In sharing sensitive data among the owners of controlled devices, it is necessary to ensure the trustworthiness of users. Although third party involvement may exist to prove valid users, the key

[1] Osaka University, Ibaraki, Osaka 567–0047, Japan
[2] Nara Institute of Science and Technology, Ikoma, Nara 630–0101, Japan
[3] National Institute of Information and Communications Technology, Koganei, Tokyo 184–8765, Japan
[a)] ei.khaing.win@ais.cmc.osaka-u.ac.jp
[b)] yoshihisa@cmc.osaka-u.ac.jp
[c)] ishi.yoshimasa@ais.cmc.osaka-u.ac.jp
[d)] kawakami@is.naist.jp
[e)] teranisi@cmc.osaka-u.ac.jp
[f)] shimojo@cmc.osaka-u.ac.jp

escrow problem should be avoided so that users can control data access directly. To be able to share the sensitive data among owners' controlled devices, certificateless public key encryption is an optimal solution.

Contribution: In this paper, we propose a lightweight and secure certificateless multi-receiver encryption scheme using elliptic curve cryptography. To ensure authentication, the proposed scheme provides implicit user authentication and source authentication. Moreover, the proposed scheme offers message integrity and replay attack prevention as the necessary security properties for secure data exchange. In addition to this, computation expensive pairing operations are eliminated to achieve less time usage for both sender and receiver. In this paper, we provide security proof for the proposed scheme based on the intractability of the Elliptic Curve Discrete Logarithm problem (ECDLP). According to the computational cost comparison and experimental results, we confirm that the proposed scheme achieves a multi-receiver encryption scheme with better efficiency and more security properties. This paper is an extension of our previous work [24]. Compared to our previous paper, this paper shows more concrete security proofs and new experimental results.

## 2. Related Works

For a multi-receiver setting, several identity-based encryption schemes and certificateless encryption schemes have been proposed.

Although identity-based encryption schemes have a key escrow problem, some identity-based encryption schemes try to avoid the key escrow problem. In identity-based multi-receiver encryption schemes (Refs. [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]), a key escrow problem exists. In Ref. [15], a key escrow problem exists although source authentication property is provided for multi-receiver setting. In an identity-based multi-receiver scheme [16], no key escrow problem exists as users generate their own key pair. Although key generation introduces some load on the sender, its merit is that there is no key escrow problem. In terms of computational cost, the scheme in Ref. [16] achieves better efficiency than the identity-based multi-receiver encryption scheme proposed in Ref. [4] because it reduces one pairing operation not only for encryption but also for decryption. However, in Ref. [16], the sender uses only the public keys of receivers for encryption. As a result, the adversary can easily impersonate the authentic receiver in communication of unknown devices in a large and highly dynamic environment. There are two options for public key validation: using the central public key directory service or explicit key validation. In the former case, communication overhead exists. Computation overhead for two pairing operations will be required in the latter case. Moreover, the scheme does not consider source authentication and replay attack prevention.

Certificateless public key scheme has been proposed to avoid the key escrow problem. However, most certificateless public key schemes are based on computation expensive bilinear pairing operations (Refs. [3], [17]). For better performance, certificateless public key encryption schemes without pairing have been proposed in Refs. [18] and [19]. To achieve a stronger security model than that of Refs. [18], [19] has been proposed. In Ref. [19], a pair of ciphertext is required for a single receiver. As a result, the ciphertext length will increase as the number of receivers becomes larger. Moreover, source authentication and replay attack prevention are not considered in both schemes. To achieve certificateless multi-receiver encryption with source authentication, Refs. [17] and [21] have been presented. In Ref. [17], expensive pairing operations are required for encryption and decryption. And the number of pairing operations is directly proportional to the number of receivers in a multi-receiver setting. Although Ref. [21] avoids a pairing operation in encryption, it requires two pairing operations for decryption and source authentication. Moreover, its ciphertext size is twice the ciphertext size of identity-based signcryption scheme [15]. To resist Type-I adversary attack in Ref. [21], an enhanced scheme is proposed in Ref. [22]. In Ref. [22], the security weakness in Ref. [21] is solved by eliminating randomness reuse in the computation of a parameter for all receivers. As a result, more pairing exponentiation is required for encryption. Moreover, decryption needs one more pairing operation and replay attack prevention is not considered. An existing certificateless multi-receiver encryption scheme has been presented in Ref. [23]. In Ref. [23], computation expensive pairing operations are used in both encryption and decryption. In addition to this, the scheme does not consider source authentication and replay attack prevention.

## 3. Preliminaries

In this section, we describe the framework of the proposed scheme, elliptic curve cryptography, computational hard problem upon which the security of the proposed scheme relies, Schnorr signature scheme and security requirements for messages.

### 3.1 Framework of the Proposed Scheme

The difference from existing scheme is that the proposed scheme eliminates computation expensive pairing operations while providing necessary security properties. We defined a multi-receiver algorithm using public-key encryptions for confidentiality, which were achieved by pairing operations in the existing schemes. Though the proposed scheme eliminates pairing operations, its confidentiality is achieved under the hard ECDLP assumption, which is proved in Section 6.1. Proposed scheme consists of five polynomial time algorithms.

- Setup: A trusted third party runs this algorithm to generate public parameters *params* and its master secret key *s*.
- Set-Key-Pair: For key escrow problem avoidance, all users (senders and receivers) run this algorithm to generate a public key and private key.
- Partial-Private-Key-Extract: A trusted third party generates identity-based partial private keys for all users based on Schnorr signature scheme.
- Encrypt: The sender runs this algorithm to encrypt the message $M$ by using the *params*, public key of a trusted third party and target receivers' public keys. To maintain message authentication and replay attack prevention properties, the sender generates a signature. Finally, ciphertext $C$ is given as output.

- Decrypt: The receiver runs this algorithm that takes $C$, *params*, partial private key and private key as input to get back plaintext message $M$.

### 3.2 Elliptic Curve Cryptography

Elliptic curve cryptography (ECC) is an approach to public-key cryptography based on the elliptic curves over finite fields. An elliptic curve $E$ over $F_p$ is defined by an equation of the form $y^2 = x^3 + ax + b \, mod \, p$ where $a, b \in F_p$ and $F_p$ is a finite field containing $p$ elements.

Let $P$ be the point on $E$. Then, the scalar multiplication $sP$ means adding $P$ for $s$-times. Elliptic curves are widely used in cryptography due to its smaller key size and faster scalar multiplication.

The security of elliptic curve cryptography is proven by the elliptic curve discrete logarithm problem (ECDLP).

### 3.3 Computational Hard Problem

Let $P$ and $Q = xP$ be two points on elliptic curve $E$. Given $P$ and $Q$, the elliptic curve discrete logarithm problem (ECDLP) is to find $x$. The size of the elliptic curve determines the difficulty of the problem.

Assumption: ECDLP is assumed to be intractable for large elliptic curve parameters.

### 3.4 Schnorr Signature Scheme

Let $G$ be a group of prime order $q$, $P$ be generator, and $H : \{0,1\}^* \in Z_q$ be one-way hash function. The Schnorr signature scheme uses a group $G$ in which the discrete logarithm problem is hard. It consists of the following algorithms.

- Key generation: Generate public key $pk = (P, Q = sP)$ by choosing private key $s \in Z_q$.
- Signing: For signing message $M \in \{0,1\}^*$, choose $r \leftarrow Z_q$ randomly. Then, compute $R = rP$, $e = H(M||R)$ and $t = r + se \, mod \, q$. The signature is $(R, t)$.
- Verifying: Compute $e(M||R)$. If $tP = R + eQ$, the signature is verified.

The security of the Schnorr signature scheme is based on the intractable one-way hash function and discrete logarithm problem.

### 3.5 Security Requirements for Message

The basic security requirements for message are message confidentiality, message authentication and replay attack prevention.

- Message Confidentiality: Confidentiality is the process that protects sensitive information or message from unauthorized receivers. Only authorized receivers can reveal the information.
- Message Authentication: Source authentication is the process that the receivers can verify the source of the message. Message integrity is the assurance that the message has not been altered in transit. Message authentication is a property that can prove both source authentication and message integrity.
- Replay Attack Prevention: A replay attack is a form of network attack in which a valid data transmission is repeated or delayed maliciously or fraudulently. It can be prevented by adding the session key or timestamp to the broadcasted message.

## 4. Security Model

In this section, we describe types of adversaries, the oracles that they can access, and the restricted behaviour for them. Then, we define the security notion for the proposed scheme: "Indistinguishability of encryptions under the adaptive chosen ciphertext attack" (IND-CCA).

### 4.1 Adversaries

We consider two types of adversaries, Type-I and Type-II adversary.

- Type-I adversary is an adversary who does not possess the master secret key. The adversary is allowed to replace the public key.
- Type-II adversary is an honest but curious *KGC* who knows the master secret key. Type-II adversary is not allowed to replace any public key.

### 4.2 Oracles for Type-I Adversary

For the security of the proposed scheme, a game is played between adversary $\mathcal{A}$ and a challenger $C$. In the game, Type-I adversary can access the following six oracles.

- CreateUser: The oracle accepts identity as input and returns the corresponding public key. If the public key for the identity does not exist in the user list, it runs Set-Key-Pair and Partial-Private-key-Extract algorithms to generate public key, private key and partial private key. And then, it returns the public key.
- Request-Public-Key : The oracle accepts an identity as input. It returns the corresponding public key.
- Replace-Public-Key: This oracle allows the adversary to replace the public key of identity with any valid values of its own choice.
- Extract-Private-Key: This oracle accepts an identity as input and outputs the corresponding private key. The restriction of the oracle is that the adversary cannot request Extract-Private-Key if he or she has already replaced the public key for identity.
- Extract-Partial-Private-Key: This oracle takes an identity as input and returns the corresponding partial private key. The restriction of the oracle is that the adversary cannot request Extract-Partial-Private-Key if he or she has already replaced the public key for identity.
- Decrypt: The oracle accepts an identity and a ciphertext as input. Using private key and partial private key corresponding to the given identity, the oracle decrypts the ciphertext and outputs the corresponding plaintext.

Type-I adversary can access all of the above oracles. The following restrictions exist for Type-I adversary.

- Adversary cannot access Extract-Partial-Private-Key oracle for the target identities and target sender at any point.
- Adversary cannot access Extract-Private-Key oracle for the target identities and target sender at any point.

### 4.3 Oracles for Type-II Adversary

Type-II adversary can access the following oracles.

- CreateUser: The oracle accepts identity as input and returns the corresponding public key. If the public key for the identity does not exist in the user list, it runs Set-Key-Pair and Partial-Private-key-Extract algorithms to generate public key, private key and partial private key. And then, it returns the public key.
- Request-Public-Key : The oracle accepts identity as input. It returns the corresponding public key.
- Extract-Private-Key: This oracle accepts an identity as input and outputs the corresponding private key. The restriction of the oracle is that the adversary cannot request Extract-Private-Key if he or she has already replaced the public key for identity.
- Extract-Partial-Private-Key: This oracle takes an identity as input and returns the corresponding partial private key. The restriction of the oracle is that the adversary cannot request Extract-Partial-Private-Key if he or she has already replaced the public key for identity.
- Decrypt: The oracle accepts an identity and a ciphertext as input. Using private key and partial private key corresponding to the given identity, the oracle decrypts the ciphertext and outputs the corresponding plaintext.

Type-II adversary can access all of the above oracles. The following restrictions exist for Type-II adversary.

- Adversary cannot access Extract-Private-Key oracle for the target identities and target sender at any point.

### 4.4 Chosen CipherText Attack (CCA)

The proposed scheme is (IND-CCA) secure if no polynomial-time adversary $\mathcal{A}$ (Type-I or Type-II) has a non-negligible advantage in the following game played against the challenger $C$ under hard ECDLP assumption.

- Setup: The challenger $C$ runs the Setup algorithm to generate public parameters *params* and master secret key $s$. It gives public parameters *params* to adversary $\mathcal{A}$. Master secret key is kept secret it if adversary $\mathcal{A}$ is Type-I adversary. In case of Type-II adversary $\mathcal{A}$, master secret key is given to $\mathcal{A}$.
- Phase 1: Challenger $C$ chooses target identities $ID_i^* = \{ID_1^*, ID_2^*, \ldots, ID_n^*\}$, target sender $ID_u$ and sends them to $\mathcal{A}$.
- Phase 2: $\mathcal{A}$ can access the following queries for $q_1, q_2, \ldots, q_{de}$ where $q_{de}$ is one of
  - Hash queries: The challenger $C$ returns the results of hashed operations for inputs given by $\mathcal{A}$.
  - Oracles: The adversary $\mathcal{A}$ can issue requests to corresponding oracles with the restricted adversary behaviour.
- Challenge: Adversary $\mathcal{A}$ chooses two plaintext messages $(M_0, M_1)$ and sends them to challenger $C$, with restriction that $M_0, M_1$ are two distinct messages of the same length. $\mathcal{A}$ is allowed for all the queries in Phase 2. $C$ then randomly selects $\alpha \in \{0, 1\}$ and ciphertext $C^*$ is generated using $ID_i^*$, $ID_u$ and $M_\alpha$.
- Phase 3: $\mathcal{A}$ can make the same queries in Phase 2, except Decrypt oracle with $C^* = C$ and $ID_i \in ID_i^*$.
- Guess: Finally, $\mathcal{A}$ outputs $\alpha' \in \{0, 1\}$ and wins the game if $\alpha' = \alpha$. The adversary $\mathcal{A}$ is defined as *IND-CCA* adversary.

The advantage of $\mathcal{A}$ to win the game is

$$Adv^{IND\text{-}CCA}(\mathcal{A}) = \left| Pr[\alpha' = \alpha] - \frac{1}{2} \right|$$

$\mathcal{A}$ breaks IND-CCA with $(ti, q_{de}, \epsilon)$ if adversary $\mathcal{A}$'s advantage $\epsilon$ is non-negligible after $q_{de}$ queries within running time $ti$. If no adversary breaks the scheme with $(ti, q_{de}, \epsilon)$, then the scheme is $(ti, q_{de}, \epsilon)$-IND-CCA secure.

## 5. Proposed Scheme

In this section, we will present the proposed scheme with security notations in detail. The security notations and their descriptions are shown in **Table 1**.

Proposed scheme consists of the following algorithms.

- Setup: Trusted third party runs this algorithm to generate public parameters *params* and master secret key ($s \in Z_q$). Master secret key is kept secret.
  $params = \{E, k0, k, P, H_1, H_2, H_3, H_4, Pub_s = sP\}$
  $H_1 : \{0,1\}^* \times G_1 \times G_1 \rightarrow Z_q, H_2 : G_1 \rightarrow \{0,1\}^{k0}, H_3 : \{0,1\}^* \rightarrow Z_q, H_4 : \{0,1\}^{k0} \rightarrow \{0,1\}^k$
- Set-Key-Pair: Each user ($i$) runs this algorithm to generate key pair. Firstly, user selects $sk_i \in Z_q$ randomly. To obtain partial private key, $(ID_i, P_i = sk_iP, P_{i1} = sk_iPub_s)$ is sent to trusted third party. After getting partial private key from trusted third party, user performs the following steps.
  - Computes $P_i' = P_i + X_i$ and $P_i'' = h_i^{-1}P_i'$ where $h_i = H_1(ID_i, P_i, P_i')$.
  - Declares $pk_i = (ID_i, P_i, P_i', P_i'')$.
- Partial-Private-Key-Extract: Trusted third party generates partial private key by performing the following steps.
  (1) Third party checks the validity of user's public key by checking whether $P_{i1}$ is equal to $sP_i$ or not. If not, third party rejects for partial private key generation.

**Table 1** Notations and descriptions.

| Notations | Descriptions |
|---|---|
| $p, q$ | Large primes |
| $E$ | Elliptic curve with domain parameters over $F_p$ |
| $P$ | Base point of order $q$ on $E$ |
| $G_1$ | Cyclic group generated by $P$ |
| $Pub_s$ | Public Key of Trusted Third Party |
| $s$ | Master Private Key of Trusted Third Party |
| *params* | Public system parameters |
| $pk_i$ | Public key of $i$-th user |
| $sk_i$ | Private key of $i$-th user |
| $X_i, d_i$ | Partial Private Key for $i$-th user |
| $_u$ | Symbol to represent the parameters of sender |
| $H_1, H_2, H_3, H_4$ | Cryptographic one-way, collision-resistant hash functions |
| $E_K$ | Symmetric encryption using key $K$ |
| $D_K$ | Symmetric decryption using key $K$ |
| $M \in \{0,1\}^*$ | Message |
| $t \in \{0,1\}^*$ | Current Time |
| $k, k_0$ | Bit lengths |

(2) Additional verification process is done using user identity and other additional verification proofs.
(3) After the user has been successfully verified, third party performs the following steps.
  – Selects $x_i \in Z_q$ uniformly at random.
  – Computes $X_i = x_iP$, $CP_i = P_i + X_i$, and $d_i = H_1(ID_i, P_i, CP_i)s + x_i \, mod \, q$.

Then, $(X_i, d_i)$ is given to user via secure channel.

- Encrypt: To encrypt message $M$ for $n$-receivers ($j = 1, 2, \ldots, n$), the sender chooses $r \in Z_q$ and $\sigma \in \{0,1\}^k$ randomly. Then, it computes

$$
\begin{aligned}
&Y = rP \\
&m = H_3(M||\sigma||L||t||Y) \\
&a = h_u^{-1}d_u + msk_u + r \, mod \, q \\
&Z = mP \\
&h_j = H_1(ID_j, P_j, P_j^{'}) \\
&U_j = mh_j(Pub_s + P_j^{''}) \\
&V = \sigma \oplus H_2(Z) \\
&K = H_4(\sigma) \\
&W = E_K(M)
\end{aligned}
$$

Then, ciphertext $C = (a, U_1, U_2, \ldots, U_n, V, W, Y, L, t)$ is sent to receivers where $L$ is a label describing association of each receiver and $U_j$.

- Decrypt: Each receiver performs the following steps to decrypt the ciphertext $C$.
(1) Find corresponding $U_j$ from label $L$.
(2) Compute $Z^{'} = (d_j + sk_j)^{-1}U_j$.
(3) Compute $\sigma^{'} = V \oplus H_2(Z^{'})$.
(4) Compute decryption key $K^{'} = H_4(\sigma^{'})$.
(5) Decrypt the ciphertext to get back the plaintext message $M^{'} = D_{K'}(W)$.
(6) Compute $m^{'} = H_3(M^{'}||\sigma^{'}||L||t||Y)$ and check whether $Y$ is equal to $aP - ((m^{'} - h_u^{-1})P_u + P_u^{''} + Pub_s)$ or not. If not, return "reject".
(7) Check whether $U_j$ and $m^{'}h_j(Pub_s + P_j^{''})$ are equal or not. If they are not equal, reject the ciphertext. Otherwise, return $M^{'}$ as the plaintext.

The correctness of the proposed scheme can be checked as follows:

$$
\begin{aligned}
Z^{'} &= (d_j + sk_j)^{-1}U_j \\
&= \frac{1}{(d_j + sk_j)}U_j \\
&= \frac{mh_j(Pub_s + P_j^{''})}{(H_1(ID_j, P_j, CP_j)s + x_j) + sk_j} \\
&= \frac{m(H_1(ID_j, P_j, P_j^{'})sP + (sk_j + x_j)P)}{H_1(ID_j, P_j, P_j^{'})s + x_j + sk_j} \\
&= \frac{m((H_1(ID_j, P_j, P_j^{'})s + sk_j + x_j)P)}{H_1(ID_j, P_j, P_j^{'})s + x_j + sk_j} \\
&= mP \\
\sigma^{'} &= V \oplus H_2(Z^{'}) \\
&= \sigma \oplus H_2(Z) \oplus H_2(Z^{'}) \\
&= \sigma \oplus H_2(mP) \oplus H_2(mP) \\
&= \sigma \\
K^{'} &= H_4(\sigma^{'}) \\
&= H_4(\sigma) \\
&= K
\end{aligned}
$$

Message authentication and replay attack prevention can be checked as follows:

$$
\begin{aligned}
h_u &= H_1(ID_u, P_u, P_u^{'}) \\
aP &= h_u^{-1}d_uP + msk_uP + rP \\
&= h_u^{-1}(h_usP + x_uP) + mP_u + rP \\
&= sP + h_u^{-1}x_uP + mP_u + rP \\
&= Pub_s + h_u^{-1}x_uP + mP_u + rP \\
(m^{'} - h_u^{-1})P_u + P_u^{''} + Pub_s \\
&= (m^{'} - h_u^{-1})P_u + h_u^{-1}(P_u + x_uP) + Pub_s \\
&= m^{'}P_u + h_u^{-1}x_uP + Pub_s \\
aP - [(m^{'} - h_u^{-1})P_u + P_u^{''} + Pub_s] \\
&= rP \\
&= Y
\end{aligned}
$$

Each user ($i$) can check the validity of third party generated partial private key by checking whether $d_iP = h_iPub_s + X_i$. Partial private key is generated based on the Schnorr signature scheme proposed in Ref. [25]. Therefore, the partial private key generation scheme using $G_1$ with hard DL assumption is provably secure in the random oracle.

## 6. Security Analysis

In this section, we show the proposed scheme is IND-CCA secure with confidentiality and provides authenticity, message integrity and replay attack prevention.

### 6.1 Security Game for Confidentiality

For the confidentiality of the proposed scheme, the security game is based on "Indistinguishability of encryptions under adaptive chosen ciphertext attacks" (IND-CCA). The proposed scheme is (IND-CCA) secure in the random oracle model if no polynomial time adversary $\mathcal{A}$ (Type-I or Type-II) has a non-negligible advantage in the following games played against the challenger $C$ under hard ECDLP assumption.

**Theorem 1:** When a polynomial time adversary $\mathcal{A}$ (Type-I adversary) can attack the scheme with advantage $\epsilon$ with the help of $H_i$ ($1 \le i \le 4$) random oracles, then there is an algorithm $\mathcal{B}$ that can solve ECDLP with non-negligible advantage. Let $h1$-list, $h2$-list, $h3$-list and $h4$-list be the result of querying $H_i$ ($1 \le i \le 4$) random oracles respectively.

**Proof:** Suppose that $\mathcal{B}$ has $(P, bP)$ as an instance of the ECDLP.

- Setup: $\mathcal{B}$ runs the setup algorithm to generate public parameters *params* where $Pub_s = bP$ and master secret key $s \leftarrow \perp$. $\mathcal{B}$ gives *params* to adversary $\mathcal{A}$.

- Phase 1: $\mathcal{B}$ outputs target identities $ID_j^* = \{ID_1^*, ID_2^*, \ldots, ID_n^*\}$, target sender $ID_u$ and send them to $\mathcal{A}$. $\mathcal{B}$ randomly picks $d_u, sk_u \in Z_q$ randomly and computes $X_u = d_uP - h_uPub_s$, $P_u = sk_uP, P_u^{'} = P_u + X_u$ and $P_u^{''} = h_u^{-1}P_u^{'}$ where $h_u = H_1(ID_u, P_u, P_u^{'})$.
- Phase 2: $\mathcal{B}$ answers several queries with restricted adversary behavior $\mathcal{A}$. Assume $L_p$ is the list of users that is initialized empty.
  - (1) CreateUser: If $(ID_i, pk_i, sk_i, D_i)$ exists in $L_p$, $\mathcal{B}$ returns $pk_i$. Otherwise, it runs the following processes.
    - (a) If $ID_i = ID_u$, then the challenger $\mathcal{B}$ sets $sk_u = \perp$ and $D_u = (X_u, d_u = \perp)$. It then returns $pk_u = (ID_u, P_u, P_u^{'}, P_u^{''})$ to $\mathcal{A}$.
    - (b) If $ID_i \in ID_j^*$, $\mathcal{B}$ picks $d_i, sk_i \in Z_q$ randomly. Then, it computes $X_i = (d_i - h_i)Pub_s$, $P_i = sk_iPub_s$, $P_i^{'} = P_i + X_i$ and $P_i^{''} = h_i^{-1}P_i^{'}$ where $h_i = H_1(ID_i, P_i, P_i^{'})$. Then, it adds $(ID_i, pk_i, sk_i, D_i)$ to $L_p$ where $pk_i = (ID_i, P_i, P_i^{'}, P_i^{''})$ and $D_i = (X_i, d_i)$. Finally, it returns $pk_i$ to adversary $\mathcal{A}$.
    - (c) If $ID_i \notin ID_j^*$, $\mathcal{B}$ picks $d_i, sk_i \in Z_q$ randomly and computes $X_i = d_iP - h_iPub_s$, $P_i = sk_iP, P_i^{'} = P_i + X_i$ and $P_i^{''} = h_i^{-1}P_i^{'}$ where $h_i = H_1(ID_i, P_i, P_i^{'})$. Then, it adds $(ID_i, pk_i, sk_i, D_i)$ to $L_p$ where $pk_i = (ID_i, P_i, P_i^{'}, P_i^{''})$ and $D_i = (X_i, d_i)$. Finally, it returns $pk_i$ to $\mathcal{A}$.
  - (2) Request-Public-Key: If $(ID_i, pk_i, sk_i, D_i)$ exists in $L_p$, $\mathcal{B}$ returns the public key $pk_i$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ makes CreateUser query.
  - (3) Replace-Public-Key: If $(ID_i, pk_i, sk_i, D_i)$ exists in $L_p$, $\mathcal{B}$ replaces the public key and set $sk_i = \perp$ and $d_i = \perp$.
  - (4) Extract-Private-Key: If $(ID_i, pk_i, sk_i, D_i)$ exists in $L_p$, $C$ outputs $sk_i$. Otherwise, $\mathcal{B}$ runs CreateUser oracle. Then, it returns $sk_i$ to $\mathcal{A}$.
  - (5) Extract-Partial-Private-Key: If $(ID_i, pk_i, sk_i, D_i)$ exists in $L_p$, $\mathcal{B}$ outputs $D_i$. Otherwise, $\mathcal{B}$ runs CreateUser oracle. Then, it returns $D_i$ to $\mathcal{A}$.
  - (6) Decrypt: Accepts $(C^*, ID_i)$ as input where $< C^* = a^*, U_1, U_2, \ldots, U_n, V^*, W^*, Y^*, L^*, t^* >$. If $(ID_i \in ID_j^*)$ or $(ID_i = ID_u)$ or $t^*$ is not within reasonable time range, return "reject". Otherwise, the process is done as follows:
    - (a) Perform Extract-Partial-Private-Key and Extract-Partial-Private oracles to get $d_i$, and $sk_i$.
    - (b) Compute $Z_i$ using $U_i$, $d_i$, and $sk_i$.
    - (c) If $(Z_i, h2_i)$ exists in $h2$-list, then compute $\sigma_i^{'} = V^* \oplus h2_i$. Otherwise, return "reject".
    - (d) If $(\sigma_i^{'}, h4_i)$ exists in $h4$-list, then decrypt message $M^{'} = D_{h4_i}(W^*)$. Otherwise, return "reject".
    - (e) If $(ID_i, P_i, P_i^{'}, h1_i)$ exists in $h1$-list and $(M^{'}||\sigma_i^{'}||L^*||t^*||Y^*, h3_i)$ in $h3$-list, then
      - check whether $Y^*$ is equal to $a^*P - ((h3_i - h_u^{-1})P_u + P_u^{''} + Pub_s)$ or not. If it is equal, continue. Otherwise, return "reject".
      - check whether $U_i$ and $h3_ih1_i(Pub_s + P_i^{''})$ are equal. If they are not equal, return "reject". Otherwise, return $M^{'}$ as the output plaintext.
- Challenge: $\mathcal{A}$ submits two plaintext messages $(M_0, M_1)$ with the same length. $\mathcal{B}$ does the encryption by randomly selecting $\alpha \in \{0,1\}$, $r \in Z_q$ and $\sigma \in \{0,1\}^k$. It then computes $< C = a, U_1, U_2, \ldots, U_n, V, W, Y, L, t >$ by performing the following steps and using the hash oracles

$$
\begin{aligned}
&Y = rP\\
&a = h_u^{-1}d_u + msk_u + r \bmod q\\
&m = H_3(M_\alpha||\sigma||L||t||Y)\\
&Z = mP\\
&h_j = H_1(ID_j, P_j, P_j^{'})\\
&U_j = mh_j(Pub_s + P_j^{''})\\
&V = \sigma \oplus H_2(Z)\\
&K = H_4(\sigma)\\
&W = E_K(M_\alpha)
\end{aligned}
$$

  Type-I adversary's random oracles $H_i$ $(1 \le i \le 4)$ work as follows:
  - $H_1$-query: Accepts $(ID_i, P_i, P_i^{'})$ as input. If $(ID_i, P_i, P_i^{'}, h1_i)$ exists in $h1$-list, then $\mathcal{B}$ returns $h1_i$. Otherwise, do the following:
    (1) Pick $h1_i \in Z_q$ randomly.
    (2) Put $(ID_i, P_i, P_i^{'}, h1_i)$ in $h1$-list.
    (3) Return $h1_i$.
  - $H_2$-query: Accepts $(Z_i)$ as input. If $(Z_i, h2_i)$ exists in $h2$-list, then $\mathcal{B}$ returns $h2_i$. Otherwise, do the following:
    (1) Pick $h2_i \in \{0,1\}^{k0}$ randomly.
    (2) Put $(Z_i, h2_i)$ in $h2$-list.
    (3) Return $h2_i$.
  - $H_3$-query: Accepts $(M_i||\sigma_i||L_i||t_i||Y_i)$ as input. If $(M_i||\sigma_i||L_i||t_i||Y_i, h3_i)$ exists in $h3$-list, then return $h3_i$. Otherwise, do the following:
    (1) Pick $h3_i \in Z_q$ randomly.
    (2) Put $(M_i||\sigma_i||L_i||t_i||Y_i, h3_i)$ in $h3$-list.
    (3) Return $h3_i$.
  - $H_4$-query: Accepts $(\sigma_i)$ as input. If $(\sigma_i, h4_i)$ exists in $h4$-list, then return $h4_i$. Otherwise, do the following:
    (1) Pick $h4_i \in \{0,1\}^k$ randomly.
    (2) Put $(\sigma_i, h4_i)$ in $h4$-list.
    (3) Return $h4_i$.
- Guess: Finally, adversary $\mathcal{A}$ outputs $\alpha^{'} \in \{0,1\}$. If $\alpha^{'} = \alpha$, challenger $\mathcal{B}$ outputs 1 and $\mathcal{A}$ wins.
- Analysis: Type-I adversary $\mathcal{A}$ can break the IND-CCA security of the proposed scheme when $\mathcal{A}$ can find $mP$ by computing $b^{-1}(d_j + sk_j)^{-1}U_j$ value. Therefore, $\mathcal{B}$ can solve ECDLP.

As discrete logarithm problem for finding $b$ is computationally intractable in polynomial time, the proposed scheme is IND-CCA secure under hard DL assumption for the elliptic curve.

Suppose the Type-I adversary can guess the value of $\alpha$ with non-negligible advantage $\epsilon$. If $H_2$ and $H_3$ are modelled as random oracles, $\mathcal{A}$ has advantage only if $mP$ is the output of $H_2$ oracle or $m$ is an output of $H_3$ oracle. The probability that the adversary can correctly guess the output of $H_2$ is $\frac{1}{2^{k0}}$. For $q_{de}$ decryption queries, adversary has advantage $\epsilon - \frac{q_{de}}{2^{k0}}$. The probability that the adversary can correctly guess the output of $H_3$ is $\frac{1}{2^q}$. For $q_{de}$ de-

cryption queries, adversary has advantage $\epsilon - \frac{q_{de}}{2^q}$. Therefore, we know that the challenger $\mathcal{B}$ can address the ECDLP problem with non-negligible advantage $\epsilon - \frac{q_{de}}{2^{k0}}$ or $\epsilon - \frac{q_{de}}{2^q}$. As discrete logarithm problem is computationally intractable in polynomial time, the proposed scheme is IND-CCA secure against Type-I adversary $\mathcal{A}$.

**Theorem 2:** When a polynomial time adversary $\mathcal{A}$ (Type-II adversary) can attack the scheme with advantage $\epsilon$ with the help of $H_i$ $(1 \leq i \leq 4)$ random oracles, then there is an algorithm $\mathcal{B}$ that can solve ECDLP with non-negligible advantage. Let $h1$-list, $h2$-list, $h3$-list and $h4$-list be the result of querying $H_i$ $(1 \leq i \leq 4)$ random oracles respectively.

**Proof:** Suppose that $\mathcal{B}$ has $(P, bP)$ as an instance of the ECDLP.

- Setup: $\mathcal{B}$ runs the setup algorithm to generate public parameters *params* where $Pub_s = sP$ and master secret key $s$. $\mathcal{B}$ gives *params* to adversary $\mathcal{A}$. Master secret key is also given to $\mathcal{A}$.
- Phase 1: $\mathcal{B}$ outputs target identities $ID_j^* = \{ID_1^*, ID_2^*, \ldots, ID_n^*\}$, target sender $ID_u$ and send them to $\mathcal{A}$. $\mathcal{B}$ picks $x_u, sk_u \in Z_q$ randomly and computes $X_u = x_u P$, $P_u = sk_u P$, $P_u^{'} = P_u + X_u$ and $P_u^{''} = h_u^{-1} P_u^{'}$ where $h_u = H_1(ID_u, P_u, P_u^{'})$. Then, it calculates $d_u = h_u s + x_u \, mod \, q$.
- Phase 2: $\mathcal{B}$ answers several queries with restricted adversary behavior $\mathcal{A}$. Assume $L_p$ is the list of users that is initialized empty.
  - (1) CreateUser: If $(ID_i, pk_i, sk_i, D_i)$ exists in $L_p$, $\mathcal{B}$ returns $pk_i$. Otherwise, it runs the following processes.
    - (a) If $ID_i = ID_u$, then the challenger $\mathcal{B}$ sets $sk_u = \perp$ and $D_u = (X_u, d_u = \perp)$. It then returns $pk_u = (ID_u, P_u, P_u^{'}, P_u^{''})$ to $\mathcal{A}$.
    - (b) If $ID_i \in ID_j^*$, $\mathcal{B}$ picks $d_i, sk_i \in Z_q$ randomly. Then, it computes $X_i = d_i(bP)$, $P_i = sk_i(bP)$, $P_i^{'} = P_i + X_i$ and $P_i^{''} = h_i^{-1} P_i^{'} - Pub_s$ where $h_i = H_1(ID_i, P_i, P_i^{'})$. And, it adds $(ID_i, pk_i, sk_i, D_i)$ to $L_p$ where $pk_i = (ID_i, P_i, P_i^{'}, P_i^{''})$ and $D_i = (X_i, d_i)$. Finally, it returns $pk_i$ to adversary $\mathcal{A}$.
    - (c) If $ID_i \notin ID_j^*$, $\mathcal{B}$ picks $x_i, sk_i \in Z_q$ randomly and computes $X_i = x_i P$, $P_i = sk_i P$, $P_i^{'} = P_i + X_i$ and $P_i^{''} = h_i^{-1} P_i^{'}$ where $h_i = H_1(ID_i, P_i, P_i^{'})$. Then, it calculates $d_i = h_i s + x_i \, mod \, q$ and adds $(ID_i, pk_i, sk_i, D_i)$ to $L_p$ where $pk_i = (ID_i, P_i, P_i^{'}, P_i^{''})$ and $D_i = (X_i, d_i)$. Finally, it returns $pk_i$ to $\mathcal{A}$.
  - (2) Request-Public-Key: If $(ID_i, pk_i, sk_i, D_i)$ exists in $L_p$, $\mathcal{B}$ returns the public key $pk_i$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ makes CreateUser query.
  - (3) Replace-Public-Key: If $(ID_i, pk_i, sk_i, D_i)$ exists in $L_p$, $\mathcal{B}$ replaces the public key and set $sk_i = \perp$ and $d_i = \perp$.
  - (4) Extract-Private-Key: If $(ID_i, pk_i, sk_i, D_i)$ exists in $L_p$, $C$ outputs $sk_i$. Otherwise, $\mathcal{B}$ runs CreateUser oracle. Then, it returns $sk_i$ to $\mathcal{A}$.
  - (5) Extract-Partial-Private-Key: If $(ID_i, pk_i, sk_i, D_i)$ exists in $L_p$, $\mathcal{B}$ outputs $D_i$. Otherwise, $\mathcal{B}$ runs CreateUser oracle. Then, it returns $D_i$ to $\mathcal{A}$.
  - (6) Decrypt: Accepts $(C^*, ID_i)$ as input where $< C^* = a^*, U_1, U_2, \ldots, U_n, V^*, W^*, Y^*, L^*, t^* >$. If $(ID_i \in ID_j^*)$ or $(ID_i = ID_u)$ or $t^*$ is not within reasonable time range, return "reject". Otherwise, the process is done as follows:
    - (a) Perform Extract-Partial-Private-Key and Extract-Partial-Private oracles to get $d_i$, and $sk_i$.
    - (b) Compute $Z_i$ using $U_i$, $d_i$, and $sk_i$.
    - (c) If $(Z_i, h2_i)$ exists in $h2$-list, then compute $\sigma_i^{'} = V^* \oplus h2_i$. Otherwise, return "reject".
    - (d) If $(\sigma_i^{'}, h4_i)$ exists in $h4$-list, then decrypt message $M^{'} = D_{h4_i}(W^*)$. Otherwise, return "reject".
    - (e) If $(ID_i, P_i, P_i^{'}, h1_i)$ exists in $h1$-list and $(M^{'} || \sigma_i^{'} || L^* || t^* || Y^*, h3_i)$ in $h3$-list, then
      - check whether $Y^*$ is equal to $a^* P - ((h3_i - h_u^{-1}) P_u + P_u^{''} + Pub_s)$ or not. If it is equal, continue. Otherwise, return "reject".
      - check whether $U_i$ and $h3_i h1_i (Pub_s + P_i^{''})$ are equal. If they are not equal, return "reject". Otherwise, return $M^{'}$ as the output plaintext.
- Challenge: $\mathcal{A}$ submits two plaintext messages $(M_0, M_1)$ with the same length. $\mathcal{B}$ does the encryption by randomly selecting $\alpha \in \{0, 1\}$, $r \in Z_q$ and $\sigma \in \{0, 1\}^k$. It then computes $< C = a, U_1, U_2, \ldots, U_n, V, W, Y, L, t >$ by performing the following steps and using the hash oracles

$$
\begin{aligned}
&Y = rP \\
&a = h_u^{-1} d_u + msk_u + r \, mod \, q \\
&m = H_3(M_\alpha || \sigma || L || t || Y) \\
&Z = mP \\
&h_j = H_1(ID_j, P_j, P_j^{'}) \\
&U_j = mh_j(Pub_s + P_j^{''}) \\
&V = \sigma \oplus H_2(Z) \\
&K = H_4(\sigma) \\
&W = E_K(M_\alpha)
\end{aligned}
$$

Type-II adversary's random oracles $H_i$ $(1 \leq i \leq 4)$ work as follows:

- $H_1$-query: Accepts $(ID_i, P_i, P_i^{'})$ as input. If $(ID_i, P_i, P_i^{'}, h1_i)$ exists in $h1$-list, then $\mathcal{B}$ returns $h1_i$. Otherwise, do the following:
  - (1) Pick $h1_i \in Z_q$ randomly.
  - (2) Put $(ID_i, P_i, P_i^{'}, h1_i)$ in $h1$-list.
  - (3) Return $h1_i$.
- $H_2$-query: Accepts $(Z_i)$ as input. If $(Z_i, h2_i)$ exists in $h2$-list, then $\mathcal{B}$ returns $h2_i$. Otherwise, do the following:
  - (1) Pick $h2_i \in \{0, 1\}^{k0}$ randomly.
  - (2) Put $(Z_i, h2_i)$ in $h2$-list.
  - (3) Return $h2_i$.
- $H_3$-query: Accepts $(M_i || \sigma_i || L_i || t_i || Y_i)$ as input. If $(M_i || \sigma_i || L_i || t_i || Y_i, h3_i)$ exists in $h3$-list, then return $h3_i$. Otherwise, do the following:
  - (1) Pick $h3_i \in Z_q$ randomly.
  - (2) Put $(M_i || \sigma_i || L_i || t_i || Y_i, h3_i)$ in $h3$-list.
  - (3) Return $h3_i$.
- $H_4$-query: Accepts $(\sigma_i)$ as input. If $(\sigma_i, h4_i)$ exists in $h4$-

list, then return $h4_i$. Otherwise, do the following:
(1) Pick $h4_i \in \{0,1\}^k$ randomly.
(2) Put $(\sigma_i, h4_i)$ in $h4$-list.
(3) Return $h4_i$.

- Guess: Finally, adversary $\mathcal{A}$ outputs $\alpha' \in \{0,1\}$. If $\alpha' = \alpha$, challenger $\mathcal{B}$ outputs 1 and $\mathcal{A}$ wins.
- Analysis: Type-II adversary $\mathcal{A}$ can break the IND-CCA security of the proposed scheme when $\mathcal{A}$ can find $mP$ by computing $b^{-1}(d_j + sk_j)^{-1}U_j$ value. Therefore, $\mathcal{B}$ can solve ECDLP.

As discrete logarithm problem for finding $b$ is computationally intractable in polynomial time, the proposed scheme is IND-CCA secure under hard DL assumption for the elliptic curve.

Suppose the Type-II adversary can guess the value of $\alpha$ with non-negligible advantage $\epsilon$. If $H_2$ and $H_3$ are modelled as random oracles, $\mathcal{A}$ has advantage only if $mP$ is the output of $H_2$ oracle or $m$ is an output of $H_3$ oracle. The probability that the adversary can correctly guess the output of $H_2$ is $\frac{1}{2^{k0}}$. For $q_{de}$ decryption queries, adversary has advantage $\epsilon - \frac{q_{de}}{2^{k0}}$. The probability that the adversary can correctly guess the output of $H_3$ is $\frac{1}{2^q}$. For $q_{de}$ decryption queries, adversary has advantage $\epsilon - \frac{q_{de}}{2^q}$. Therefore, we know that the challenger $\mathcal{B}$ can address the ECDLP problem with non-negligible advantage $\epsilon - \frac{q_{de}}{2^{k0}}$ or $\epsilon - \frac{q_{de}}{2^q}$. As discrete logarithm problem is computationally intractable in polynomial time, the proposed scheme is IND-CCA secure against Type-II adversary $\mathcal{A}$.

### 6.2 Source Authentication

**Theorem 3:** Under hard discrete logarithm assumption, the signature $(a)$ is existentially unforgeable in the random oracle model.

**Proof:** Let $\mathcal{A}$ be any probabilistic time adversary with running time $t_A$, making $q_s$ queries to signature oracle, and $q_{H3}$ random oracle queries to $H_3$ oracle. $\mathcal{B}$ acts as a challenger and responds to $\mathcal{A}$'s signature and $H_3$ queries. Public key of third party, public key of target sender $(pk_u)$ and $X_u$ are given to challenger $\mathcal{B}$ as the signature public keys.

- Challenger $\mathcal{B}$ sends the signature public keys to $\mathcal{A}$. $\mathcal{B}$ then chooses $w \in [1, q_{H3}]$ randomly. Assume that the adversary $\mathcal{A}$ will forge the signature on the $w$th $H_3$ query.
- signature oracle: For i-th signature query using message $(M_i, \sigma_i, L_i, t_i)$, $\mathcal{B}$ does the following
  (1) Choose random $a_i, m_i$ from $Z_q$.
  (2) Set $Y_i = a_iP - m_iP_u - h_u^{-1}(h_uPub_s + X_u)$.
  (3) return $Y_i, a_i$.
  (4) Store $m_i$ as the value of $H_3(M_i||\sigma_i||L_i||t_i||Y_i)$.
- For $j$th $H_3$ query on $(M_j||\sigma_j||L_j||t_j||Y_j)$, $\mathcal{B}$ does the following
  (1) If the value of $H_3$ query already exists, return the value.
  (2) If $j \neq w$, choose $m_j \leftarrow Z_q$ and set $m_j$ as the result of $H_3$ query on $(M_j||\sigma_j||L_j||t_j||Y_j)$.
  (3) If $j = w$, send $Y_j$ to valid receiver of target sender and obtain a challenge $m^*$ from that receiver. Then, hash value of $H_3(M_j||\sigma_j||L_j||t_j||Y_j)$ is set as $m^*$.
- On receiving a forgery attempt $(M^*, \sigma^*, L^*, t^*, Y^*, a^*)$ from $\mathcal{A}$, send $a^*$ to valid receiver of target sender. If the $j$th hash query is $(M^*||\sigma^*||L^*||t^*||Y^*)$ and $a^*P = Y^* + m^*P_u - h_u^{-1}(h_uPub_s + X_u)$. Then, the signature forgery is valid and the source authentication property of the proposed scheme is broken.

The probability that the $j$th hash query gets valid input will be $1/(q_{H3})$. And the probability that signature oracle issues duplicate $Y$ value is $(q_{H3}+q_S+1)/q$. Suppose the probability that adversary $\mathcal{A}$ can produce a valid signature forgery is $\epsilon$. Then, there is an algorithm $\mathcal{B}$ that can impersonate the valid sender with probability at least $\epsilon/(q_{H3}) - (q_{H3} + q_S + 1)/q$.

To forge the valid signature $a$, the third party needs to reveal the value of $sk_u$ from $P_u$. Suppose trusted third party calculates $h_u^{-1}d_u$ and $a - h_u^{-1}d_u$. Then, the result $msk_u + r$ is same to the Schnorr signature scheme. Therefore, the security of the source authentication lies on the ECDLP problem that is computationally intractable in polynomial time. As a result, the signature $a$ is existentially unforgeable in the random oracle model.

### 6.3 Implicit User Authentication

**Theorem 4:** According to the security model for adversaries and the discrete logarithm assumption holds in the elliptic group, then it also provides implicit user authentication in the random oracle model with the restricted adversary behaviour.

**Proof:** For implicit user authentication, the sender uses the public key of receivers and public key of a trusted third party. Therefore, only receivers who have a corresponding private key and master secret key of third party can decrypt the ciphertext. Receivers knows the master secret key of the third party indirectly from the valid partial private key. Among the oracles for a different adversary, the following two oracles also exist.

- Extract-Private-Key: This oracle accepts an identity as input and outputs the corresponding private key. The restriction of the oracle is that the adversary cannot request Extract-Private-Key if he or she has already replaced the public key for identity.
- Extract-Partial-Private-Key: This oracle takes an identity as input and returns the corresponding partial private key. The restriction of the oracle is that the adversary cannot request Extract-Partial-Private-Key if he or she has already replaced the public key for identity.

However, according to the security model, the adversary (Type-I) cannot access Extract-Partial-Private-Key and Extract-Private-Key random oracles for the target identities and target sender at any point. And a Type-II adversary cannot access Extract-Private-Key oracle for the target identities and target sender at any point. Therefore, finding the private key of the receiver from its public key is the elliptic curve discrete problem. Therefore, the proposed scheme implicitly achieves user authentication.

### 6.4 Message Integrity

**Theorem 5:** For the message integrity, we consider the game played between polynomial time adversary $\mathcal{A}$ and challenger $\mathcal{B}$. The proposed scheme is secure against ciphertext forgery in the random oracle model if no polynomially-bounded adversary has a non-negligible advantage in the following game. The game uses the same random oracles $H_i$ $(1 \leq i \leq 4)$ described in Section 6.1.

- $\mathcal{B}$ runs the setup algorithm to generate public parameters *params* where $Pub_s = sP$ and master secret key $s$. $\mathcal{B}$ gives *params* to adversary $\mathcal{A}$.
- Phase 1: $\mathcal{B}$ outputs target identities $ID_j^* = \{ID_1^*, ID_2^*, \ldots, ID_n^*\}$, target sender $ID_u$ and send them to $\mathcal{A}$.
- Phase 2: $\mathcal{B}$ answers several queries with restricted adversary behavior $\mathcal{A}$. Assume $L_p$ is the list of users that is initialized empty.
  (1) CreateUser: If $(ID_i, pk_i, sk_i, D_i)$ exists in $L_p$, $\mathcal{B}$ returns $pk_i$. Otherwise, it runs the following processes.
    (a) If $ID_i = ID_u$ or $ID_i \in ID_j^*$, $\mathcal{B}$ picks $sk_i \in Z_q$ randomly. Then, it computes $X_i = bP$, $P_i = sk_iP$, $P_i^{'} = P_i$ and $P_i^{''} = h_i^{-1}P_i^{'}$ where $h_i = H_1(ID_i, P_i, P_i^{'})$. Then, it assigns $sk_i = \perp$ and $d_i = \perp$. Then, it adds $(ID_i, pk_i, sk_i, D_i)$ to $L_p$ where $pk_i = (ID_i, P_i, P_i^{'}, P_i^{''})$ and $D_i = (X_i, d_i)$. Finally, it returns $pk_i$ to adversary $\mathcal{A}$.
    (b) If $ID_i \notin ID_j^*$, $\mathcal{B}$ picks $x_i, sk_i \in Z_q$ randomly and computes $X_i = x_iP$, $P_i = sk_iP, P_i^{'} = P_i + X_i$ and $P_i^{''} = h_i^{-1}P_i^{'}$ where $h_i = H_1(ID_i, P_i, P_i^{'})$. Then, it performs $d_i = h_is + x_i \, mod \, q$. Then, it adds $(ID_i, pk_i, sk_i, D_i)$ to $L_p$ where $pk_i = (ID_i, P_i, P_i^{'}, P_i^{''})$ and $D_i = (X_i, d_i)$. Finally, it returns $pk_i$ to $\mathcal{A}$.
  (2) Request-Public-Key: If $(ID_i, pk_i, sk_i, D_i)$ exists in $L_p$, $\mathcal{B}$ returns the public key $pk_i$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ makes CreateUser query.
  (3) Replace-Public-Key: If $(ID_i, pk_i, sk_i, D_i)$ exists in $L_p$, $\mathcal{B}$ replaces the public key and set $sk_i = \perp$ and $d_i = \perp$.
  (4) Extract-Private-Key: If $(ID_i, pk_i, sk_i, D_i)$ exists in $L_p$, $C$ outputs $sk_i$. Otherwise, $\mathcal{B}$ runs CreateUser oracle. Then, it returns $sk_i$ to $\mathcal{A}$.
  (5) Extract-Partial-Private-Key: If $(ID_i, pk_i, sk_i, D_i)$ exists in $L_p$, $\mathcal{B}$ outputs $D_i$. Otherwise, $\mathcal{B}$ runs CreateUser oracle. Then, it returns $D_i$ to $\mathcal{A}$.
  (6) Decrypt: Accepts $(C^*, ID_i)$ as input where $< C^* = a^*, U_1, U_2, .., U_n, V^*, W^*, Y^*, L^*, t^* >$. If $(ID_i \in ID_j^*)$ or $(ID_i = ID_u)$ or $t^*$ is not within reasonable time range, return "reject". Otherwise, the process is done as follows:
    (a) Perform Extract-Partial-Private-Key and Extract-Partial-Private oracles to get $d_i$, and $sk_i$.
    (b) Compute $Z_i$ using $U_i$, $d_i$, and $sk_i$.
    (c) If $(Z_i, h2_i)$ exists in $h2$-list, then compute $\sigma_i^{'} = V^* \oplus h2_i$. Otherwise, return "reject".
    (d) If $(\sigma_i^{'}, h4_i)$ exists in $h4$-list, then decrypt message $M^{'} = D_{h4_i}(W^*)$. Otherwise, return "reject".
    (e) If $(ID_i, P_i, P_i^{'}, h1_i)$ exists in $h1$-list and $(M^{'} || \sigma_i^{'} || L^* || t^* || Y^*, h3_i)$ in $h3$-list, then
      – check whether $Y^*$ is equal to $a^*P - ((h3_i - h_u^{-1})P_u + P_u^{''} + Pub_s)$ or not. If it is equal, continue. Otherwise, return "reject".
      – check whether $U_i$ and $h3_ih1_i(Pub_s + P_i^{''})$ are equal. If they are not equal, return "reject". Otherwise, return $M^{'}$ as the output plaintext.
- Challenge: Adversary $\mathcal{A}$ performs a number of queries described above to output a valid ciphertext from $ID_u$ to $ID_j^*$.
- Guess: Finally, adversary $\mathcal{A}$ outputs ciphertext from $ID_u$ to $ID_j^*$. If the ciphertext is valid, challenger $\mathcal{B}$ outputs 1 and $\mathcal{A}$ wins.
- Analysis: Suppose the adversary can forge the ciphertext with non-negligible advantage $\epsilon$. If $H_3$ is modelled as random oracle, $\mathcal{A}$ has advantage only if $m$ is an output of $H_3$ oracle. The probability that the adversary can correctly guess the output of $H_3$ is $\frac{1}{2^q}$. However, if signature unforgeability exists under hard discrete logarithm assumption, the proposed scheme is secure against the ciphertext forgery in the random oracle.

### 6.5 Replay Attack Prevention

**Theorem 6:** If the proposed scheme is secure against signature forgery if no polynomially bounded adversary has non-negligible advantage, then the proposed scheme is secure against replay attack in the random oracle model.

**Proof:** In our proposed scheme, timestamp ($t$) is added in calculation of $m$ value. Again, ($m$) is used to create signature ($a$). Although timestamp value is public, replay attack is prevented as the signature $a$ is existentially unforgeable in the random oracle model. So, the proposed scheme is secure against replay attack in the random oracle model since it is secure against signature forgery in the random oracle model if no polynomially bounded adversary has a non-negligible advantage.

To use timestamps for preventing replay attacks, the synchronization of time at the sender and receiver is required. The validity of timestamp should be restricted to a short time period in order to tolerate data delivery latency or time inaccuracy. Time synchronization can be done by using several time synchronization methods such as Network Time Protocol (NTP).

## 7. Performance Analysis

In this section, experimental performance evaluations of the proposed scheme were done to show the computational cost comparisons with the existing schemes and the feasibility of the actual IoT scenarios.

### 7.1 Assumed IoT Scenario

In the many IoT systems, to process data that is generated by lower-performance embedded devices, they assume computationally powerful and network-connected computers such as smartphones, home computers, and so on that work as gateways for such devices. The gateway has the computation ability to process data instead of processing data on the lower-performance devices. This system architecture is the so-called "edge computing environment" [26], [27]. In the performance evaluation, we also assume such architecture to treat secure multi-receiver data delivery with encryptions for lower-performance devices. The gateway encrypts and decrypts data using our proposed scheme for multi-receiver data delivery on behalf of the lower-performance devices. The lower-performance devices only need to connect to a gateway using standard lightweight secure protocols such as Bluetooth and WiFi. In the evaluation, we measured the encryption and decryption performances on the gateways. We used a

smart phone with android version 4.4.2 with Quad-core 1.2 GHz Cortex-A9 CPU and 1 GB RAM as an average performance gateway. We assumed a desirable encryption or decryption time on the gateway device as less than 10 seconds, which is a tolerable response time for many applications [28], [29]. The performance of network transfer latency and overheads on lower-performance devices are beyond the scope of this evaluation.

### 7.2 Experimental Setup

For existing schemes [16], [22], [23], we use a super singular elliptic curve over $F_p$ having subgroup of order $q$ by defining values of $p = 512$ bits, $q = 160$ bits, and length of $k0 = k = 192$ bits. For pairing operation, symmetric bilinear pairing $e : G_1 \times G_1 \rightarrow G_2$ is used where $G_1$ is an additive cyclic group of prime order $q$ and $G_2$ is a multiplicative cyclic group of prime order $q$. Implementation is done using android library 5.1.1 and jpbc 1.2.1 library [30]. To implement the same security level, in the proposed scheme, we use a non-singular elliptic curve over $F_p$ having group of order $q$ where $p = 160$ bits and $q = 160$ bits, and bouncy castle library [31]. To compare computation cost, operation costs for elements in $G_1$ and $G_2$ are considered. Notations and descriptions for operations are described in **Table 2**. Although the proposed scheme requires one more scalar multiplication for public key $P_i^{''}$, it can be neglected as it is calculated just for once.

### 7.3 Security Properties

Although all schemes avoid the key escrow problem, only the proposed scheme provides replay attack prevention. Among all schemes, the scheme described in Ref. [22] and the proposed scheme achieve source authentication. In existing scheme [16], implicit user authentication property fails as the sender uses the public keys of receivers in encryption. Without checking the validity and authenticity of public keys, adversaries can easily impersonate authentic receivers. To authenticate the receivers, the scheme will require two pairing operations. From the perspective of security properties, the schemes are compared in **Table 3**.

**Table 2** Notations and descriptions for point operation.

| Notations | Description |
|---|---|
| $T_{pair}$ | Time cost for Pairing operation |
| $T_{mul}$ | Time cost for Scalar multiplication |
| $T_{pow}$ | Time cost for Exponentiation in $G_1$ |
| $T_{add}$ | Time cost for Addition in $G_1$ |
| $T_{expo}$ | Time cost for Exponentiation in $G_2$ |
| $T_{pdiv}$ | Time cost for Pairing division |

**Table 3** Security properties comparison.

| | Ref. [16] | Ref. [22] | Ref. [23] | Proposed Scheme |
|---|---|---|---|---|
| Key Escrow Problem | No | No | No | No |
| Message Integrity | Yes | Yes | Yes | Yes |
| Source Authentication | No | Yes | No | Yes |
| Replay Attack Prevention | No | No | No | Yes |
| Implicit User Authentication | No | Yes | Yes | Yes |

### 7.4 Encryption Cost

In existing scheme [16], the encryption cost of sender for $n$-receivers is $n(2T_{mul} + T_{add}) + T_{expo}$. For $n$-receivers, encryption time cost of sender by eliminating pre-computed pairing operation is $n(T_{mul} + T_{expo} + T_{pow}) + 2T_{mul}$ and $n(2T_{mul} + T_{add}) + 3T_{mul}$ for Refs. [22] and [23] respectively. In Ref. [22], the encryption cost will be $n(2T_{mul} + T_{expo} + T_{pow} + T_{add}) + 2T_{mul}$ if the sender is not PKG that knows the master secret key. In the proposed scheme, $n(T_{mul} + T_{add}) + 2T_{mul}$ is required for $n$-receivers encryption. Computational cost for encryption is compared in **Table 4**.

Experimental comparison of encryption time is shown in **Fig. 1**. Among existing schemes, Ref. [16] requires less encryption time cost than that of Refs. [22] and [23]. That is because the computational cost is less than that of Refs. [22] and [23]. Although additional security properties are provided, the proposed scheme achieves faster encryption time for multi-receivers than all of the existing schemes because of less computational cost for encryption.

As the evaluation result shows, the encryption time is proportional to the number of receivers. Therefore, if the number of receivers becomes larger, the proposed scheme may not satisfy the system requirements. According to the evaluation result in Fig. 1, it takes longer than 10 seconds to encrypt data when there are 30 receivers in the proposed scheme. In the existing schemes, it took 15–22 seconds when there are 30 receivers. To achieve lower encryption time for a large number of receivers, we can apply the proposed scheme for initial key delivery for later communication via NIST-approved lightweight cryptographic primitives [32].

### 7.5 Decryption Cost

We consider the decryption time cost for a receiver. The scheme in Ref. [16] requires $T_{pair} + T_{expo}$ for calculating $\sigma^{'}$ and $2T_{mul} + T_{add}$ for checking message integrity and $U_i$ con-

**Table 4** Encryption cost comparison.

| | Encryption Cost for (n-receiver) | | | |
|---|---|---|---|---|
| | Ref. [16] | Ref. [22] | Ref. [23] | Proposed Scheme |
| $T_{pair}$ | 0 | 0 (or) 1 | 0 (or) 1 | 0 |
| $T_{mul}$ | 2n | n+2 | 2n+3 | n+2 |
| $T_{pow}$ | 0 | n | 0 | 0 |
| $T_{add}$ | n | 0 | n | n |
| $T_{expo}$ | 1 | n | 0 | 0 |

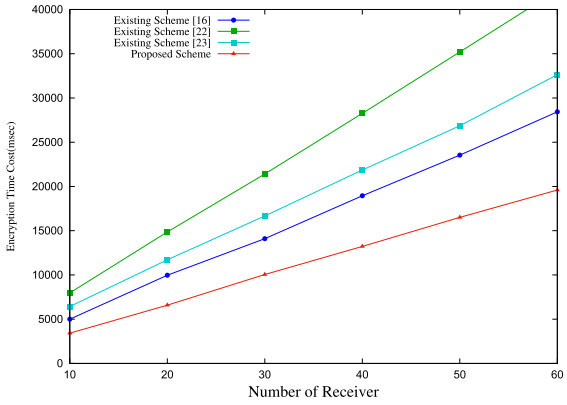


**Fig. 1** Experimental comparison of encryption time.

**Table 5** Decryption cost comparison.

| | Decryption Cost for 1-receiver | | | |
|---|---|---|---|---|
| | Ref. [16] | Ref. [22] | Ref. [23] | Proposed Scheme |
| $T_{pair}$ | 1 | 3 | 4 | 0 |
| $T_{mul}$ | 2 | 2 | 0 | 4 |
| $T_{add}$ | 1 | 2 | 0 | 4 |
| $T_{expo}$ | 1 | 1 | 0 | 0 |
| $T_{pdiv}$ | 0 | 0 | 1 | 0 |

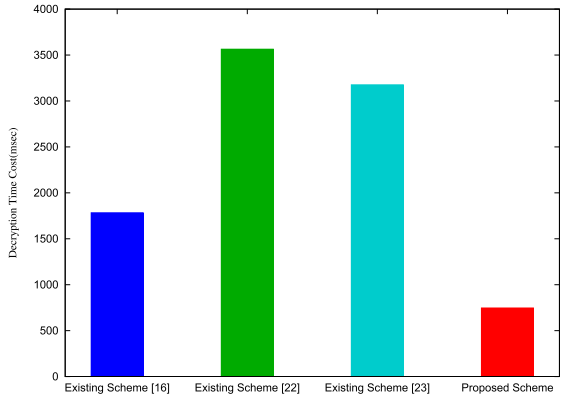


**Fig. 2** Experimental comparison of decryption time.

sistency. The scheme [22] requires $T_{pair} + T_{expo}$ for decryption and $2T_{pair} + 2T_{mul} + 2T_{add}$ for checking message authentication. In Ref. [23], decryption cost is $4T_{pair} + T_{pdiv}$. In the proposed scheme, decryption time cost is $T_{mul}$ for calculating $\sigma'$, $T_{mul}$ for $U_i$ consistency, and $2T_{mul} + 4T_{add}$ for checking message authentication and replay attack. Computational cost for decryption is compared in **Table 5**.

Experimental comparison of decryption time is shown in **Fig. 2**. Among all schemes, the decryption cost of Ref. [22] is the highest as it uses more computation expensive pairing operations. The decryption cost of Ref. [16] is less than that of Refs. [22] and [23] as it uses less pairing operation. Among all schemes, the proposed scheme achieves the fastest decryption cost because it avoids the computation of expensive pairing operations in decryption.

## 8. Concluding Remarks

We have proposed an efficient and secure multi-receiver encryption scheme with lightweight nature for the device to device communications on Internet of Things (IoT) applications. Our proposed scheme avoids the inherent key escrow problem as existing certificate-based and certificateless multi-receiver encryption schemes do. Moreover, our proposed scheme achieves multi-receiver encryption with better efficiency and more security properties. Under Discrete Logarithm assumption, security proofs in the random oracles are also given for the proposed scheme. From the computational cost comparison and experimental results, the proposed scheme achieves faster encryption and decryption time compared to the existing schemes in a multi-receiver environment.

**Acknowledgments** This work was supported in part by JSPS KAKENHI Grant Numbers JP15H02702, JP17K00146, and JP18K11316. Also this work includes the result of NICT Osaka University joint research "research and development of advanced network platform technology for large scale distributed computing".

## References

[1] Gutmann, P.: PKI: It's not dead, just resting, *J. Computer*, Vol.35, No.8, pp.41–49 (2002).
[2] Shamir, A.: Identity-based cryptosystems and signature schemes, *Proc. Intl. Workshop. Theory and Application of Cryptographic Techniques*, pp.47–53, Springer (1984).
[3] Al-Riyami, S.S. and Paterson, K.G.: Certificateless public key cryptography, *Proc. Intl. Conf. Theory and Application of Cryptology and Information Security*, pp.452–473, Springer (2003).
[4] Baek, J., Safavi-Naini, R. and Susilo, W.: Efficient multi-receiver identity-based encryption and its application to broadcast encryption, *Proc. Intl. Workshop. Public Key Cryptography*, pp.380–397, Springer (2005).
[5] Delerablée, C.: Identity-based broadcast encryption with constant size ciphertexts and private keys, *Proc. Intl. Conf. Theory and Application of Cryptology and Information Security*, pp.200–215, Springer (2007).
[6] Sakai, R. and Furukawa, J.: Identity-Based Broadcast Encryption, *J. IACR Cryptology ePrint Archive*, Vol.2007, p.217 (2007).
[7] Jiang, H., Xu, Q. and Shang, J.: An efficient dynamic identity-based broadcast encryption scheme, *Proc. Intl. Symposium. Data, Privacy and E-Commerce* (*ISDPE*), pp.27–32, IEEE (2010).
[8] Hu, L., Liu, Z. and Cheng, X.: Efficient Identity-based broadcast encryption without random oracles, *J. Computers*, Vol.5, No.3, pp.331–336 (2010).
[9] Ming, Y. and Shen, X.: Multi-receiver Identity-Based Key Encapsulation in the Standard Model, *Proc. Intl. Conf. Information Science and Management Engineering*, Vol.1, pp.382–385, IEEE (2010).
[10] Fan, C., Huang, L. and Ho, P.: Anonymous multireceiver identity-based encryption, *Trans. Computers*, Vol.59, No.9, pp.1239–1249, IEEE (2010).
[11] Zhang, J. and Mao, J.: An improved anonymous multireceiver identity-based encryption scheme, *J. Communication Systems*, Wiley Online Library, Vol.28, No.4, pp.645–658 (2015).
[12] Fan, C. and Tseng, Y.: Anonymous Multi-Receiver Identity-Based Authenticated Encryption with CCA Security, *J. Symmetry*, Vol.7, No.4, pp.1856–1881, Multidisciplinary Digital Publishing Institute (2015).
[13] Wang, S.: Practical identity-based encryption (IBE) in multiple PKG environments and its applications, arXiv prepring cs/0703106, (2007).
[14] Qin, L., Cao, Z. and Dong, X.: Multi-Receiver identity-based encryption in multiple PKG environment, *Proc. Intl. Conf. Global Communications Conference*, pp.1–5, IEEE GLOBECOM (2008).
[15] Selvi, S.S.D., Vivek, S.S., et al.: An efficient identity-based signcryption scheme for multiple receivers, *Proc. Intl. Workshop on Security*, pp.71–88, Springer (2009).
[16] Sur, C., Jung, C.D. and Rhee, K.-H.: Multi-receiver certificate-based encryption and application to public key broadcast encryption, *Proc. Symposium Bio-inspired, Learning, and Intelligent Systems for Security*, pp.35–40, IEEE (2007).
[17] Lee, Y.-R. and Lee, H.-S.: An authenticated certificateless public key encryption scheme, *Trends in Mathematics Information Center for Mathematical Sciences*, Vol.8, No.1, pp.177–187 (2005).
[18] Baek, J., Safavi-Naini, R. and Susilo, W.: Certificate-less public key encryption without pairing, *Proc. Intl. Conf. Information Security*, pp.134–148, Springer (2005).
[19] Sun, Y., Zhang, F. and Baek, J.: Strongly secure certificateless public key encryption without pairing, *Proc. Intl. Conf. Cryptology and Network Security*, pp.194–208, Springer (2007).
[20] Choi, K.Y., Park, J.H., Hwang, J.Y. and Lee, D.H.: Efficient certificateless signature schemes, *Applied Cryptography and Network Security*, pp.443–458, Springer (2007).
[21] Selvi, S., Vivek, S., Shukla, D. and Chandrasekaran, P.: Efficient and provably secure certificateless multi-receiver signcryption, *Proc. Intl. Conf. Provable Security*, pp.52–67, Springer (2008).
[22] Selvi, S.S.D., Vivek, S.S. and Rangan, C.P.: A note on the Certificateless Multi-receiver Signcryption Scheme, *Proc. J. IACR Cryptology ePrint Archive*, Vol.2009 (2009).
[23] Zhu, J., Chen, L.L., Zhu, X. and Xie, L.: A new efficient certificateless multi-receiver public key encryption scheme, *J. Computer Science Issues*, Vol.13, No.6, pp.1–7 (2016).
[24] Win, E.K., Yoshihisa, T., Ishi, Y., Kawakami, T., Teranishi, Y. and Shimojo, S.: A Lightweight Multi-receiver Encryption Scheme with Mutual Authentication, *Proc. Intl. Conf. Computer Software and Applications* (*COMPSAC*), Vol.2, pp.491–497 (2017).

[25] Freeman, D.M.: Schnorr Identification and Signatures (2010).
[26] Tanaka, H., Yoshida, M., Mori, K. and Takahashi, N.: Multi-access Edge Computing: A Survey, *J. Information Processing*, Vol.26, pp.87–97 (2018).
[27] Shi, W., Cao, J., Zhang, Q., Li, Y. and Xu, L.: Edge computing: Vision and challenges, *IEEE Internet of Things J.*, Vol.3, No.5, pp.637–646 (2016).
[28] Nielsen, J.: Usability engineering, Elsevier (1994).
[29] Hoxmeier, J.A. and DiCesare, C.: System response time and user satisfaction: An experimental study of browser-based applications, *Proc. AMCIS 2000* (2000).
[30] De Caro, A. and Iovino, V.: jPBC: Java pairing based cryptography, *Proc. 16th IEEE Symposium on Computers and Communications*, pp.850–855, IEEE (2011).
[31] Bouncy Castle, available from ⟨http://www.bouncycastle.org/⟩.
[32] McKay, K.A., Bassham, L., Turan, M.S. and Mouha, N.: Report on lightweight cryptography, *NIST DRAFT NISTIR J.*, Vol.8114 (2016).

**Ei Khaing Win** received her B.C.Sc. degree from University of Computer Studies (Mandalay), Myanmar, in 2004, her M.C.Sc. degree from Computer University (Mandalay), Myanmar, in 2010, and Doctor's degree from Osaka University, Osaka, in 2018, respectively. From 2007 to 2009, she was appointed as a demonstrator of Computer University (Hinthada), Myanmar. From 2009 to 2015, she became a tutor of University of Technology (Yatanarpon Cyber City), Myanmar. Since October 2015, she has been an assistant lecturer of University of Technology (Yatanarpon Cyber City), Myanmar. Her research interests include security and stream data processing.

**Tomoki Yoshihisa** received his Bachelor's, Master's, and Doctor's degrees from Osaka University, Osaka, Japan, in 2002, 2003, 2005, respectively. Since 2005 to 2007, he was a research associate at Kyoto University. In January 2008, he joined the Cybermedia Center, Osaka University as an assistant professor and in March 2009, he became an associate professor. From April 2008 to August 2008, he was a visiting researcher at University of California, Irvine. His research interests include video-on-demand, broadcasting systems, and webcasts. He is a member of the IEICE and IEEE.

**Yoshimasa Ishi** received his B.E. degrees from Kyoto Institute of Technology, Japan, in 2004 and his M.I. degrees from Osaka University, Japan, in 2006, respectively. From 2006 to 2008, and from 2012 to 2015, he was a specially appointed researcher of Cybermedia Center, Osaka University. From 2008 to 2012, he was a specially appointed researcher of Graduate School of Information Science and Technology, Osaka University. Since January 2017, he has been a specially appointed researcher of Institute for Datability Science, Osaka University. His research interests include technologies for distributed network systems and its development.

**Tomoya Kawakami** received his B.E. degree from Kinki University in 2005 and his M.I. and Ph.D. degrees from Osaka University in 2007 and 2013, respectively. From 2007 to March 2013 and from July 2014 to March 2015, he was a specially appointed researcher at Osaka University. From April 2013 to June 2014, he was a Ph.D. researcher at Kobe University. Since April 2015, he has been an assistant professor at Nara Institute of Science and Technology. His research interests include distributed computing, rule-based systems, and stream data processing. He is a member of the IEEE.

**Yuuichi Teranishi** received his M.E. and Ph.D. degrees from Osaka University, Japan, in 1995 and 2004, respectively. From 1995 to 2004, he was engaged Nippon Telegraph and Telephone Corporation (NTT). From 2005 to 2007, he was a Lecturer of Cybermedia Center, Osaka University. From 2007 to 2011, he was an associate professor of Graduate School of Information Science and Technology, Osaka University. Since August 2011, he has been a research manager and project manager of National Institute of Information and Communications Technology (NICT). He received IPSJ Best Paper Award in 2011. His research interests include technologies for distributed network systems and applications. He is a member of the IEICE and IEEE.

**Shinji Shimojo** received his M.E. and Ph.D. degrees from Osaka University in 1983 and 1986, respectively. He was an Assistant Professor with the Department of Information and Computer Sciences, Faculty of Engineering Science at Osaka University from 1986, and an Associate Professor with Computation Center from 1991 to 1998. During this period, he also worked for a year as a Visiting Researcher at the University of California, Irvine. He has been a Professor with the Cybermedia Center (then the Computation Center) at Osaka University since 1998, and from 2005 to 2008, and since 2016, he had/has been the director of the Center. He is an executive researcher at National Institute of Information and Communications Technology and a director of Network Testbed Research and Development Promotion Center. His current research work is focusing on a wide variety of multimedia applications, peer-to-peer communication networks, ubiquitous network systems, and grid technologies. He was awarded the Osaka Science Prize in 2005. He is a member of IEEE and IEICE.